Science & Technology
Facilities Council

The ICAT pluggable authentication system

Tom Griffin, STFC ISIS Facility

NOBUGS 2012 ICAT Workshop

tom.griffin@stfc.ac.uk

History

- ICAT 3 – compile time changeable user single plug-in
- ICAT 4.0/4.1 – deploy time changeable single plug-in
- ICAT 4.2 – new ‘Authenticator’ interface. New login method signature. Multiple plugins. ‘Live’ configurable. IP address parameter

3.3 ‘Login’ method

```
String sessionId = port.login(“tom", “mypassword");
```

4.2 ‘Login’ method

~~String sessionId = port.login(“tom”, “mypassword”);~~

```
Login.Credentials.Entry usernameEntry = new Login.Credentials.Entry();
usernameEntry.setKey("username");
usernameEntry.setValue("tom");
Login.Credentials.Entry passwordEntry = new Login.Credentials.Entry();
passwordEntry.setKey("password");
passwordEntry.setValue(“mypassword");
Login.Credentials creds = new Login.Credentials();
creds.getEntry().add(usernameEntry);
creds.getEntry().add(passwordEntry);
String sessionId = port.login("db", creds);
```

4.2 ‘Login’ method

```
Login.Credentials.Entry certEntry = new Login.Credentials.Entry();
usernameEntry.setKey(“certificate");
usernameEntry.setValue(“2485485464251654548786546548746548 7…");
Login.Credentials creds = new Login.Credentials();
creds.getEntry().add(certEntry);
String sessionId = port.login(“certificate", creds);
```

Science & Technology Facilities Council

4.2 ‘Login’ method

```
Login.Credentials.Entry certEntry = new Login.Credentials.Entry();
usernameEntry.setKey(“certificate");
usernameEntry.setValue(“2485485464251654548786546548746548 7…");
Login.Credentials creds = new Login.Credentials();
creds.getEntry().add(certEntry);
String sessionId = port.login(“certificate", creds);
```

Science & Technology Facilities Council

Authenticator Interface

```
public interface Authenticator {
  Authentication authenticate(Map<String, String> credentials,
String remoteAddr)
          throws IcatException;
}
```

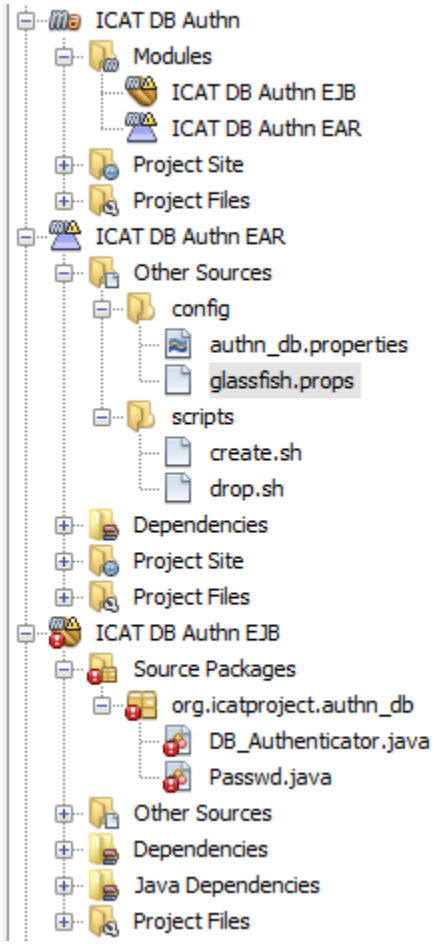
Also provides ‘AddressChecker’

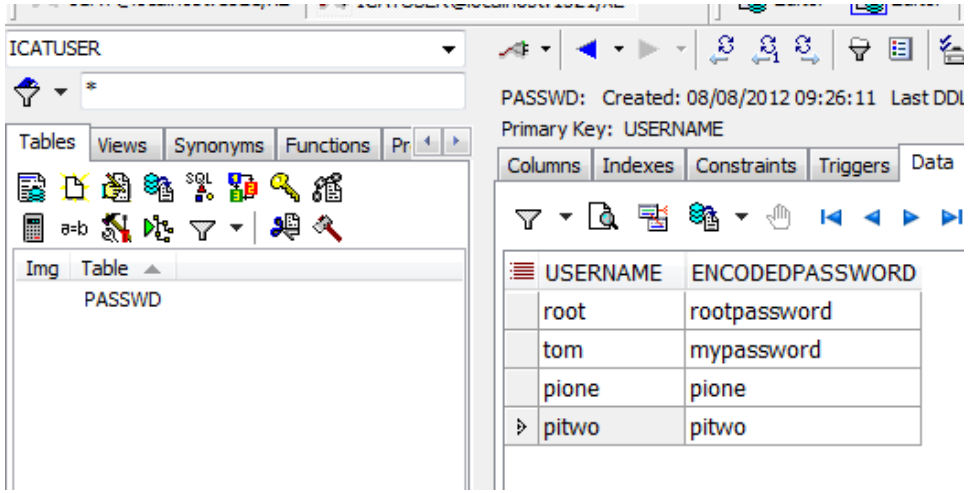
Validates calling IP address against a pattern

Can be optionally called by Authenticator implementation

Example – ‘db’

Simple classes, packages as JAR then EAR





Science & Technology Facilities Council

Example – ‘db’

```
@PostConstruct
   private void init() {
      File f = new File("authn_db.properties");
      Properties props = null;
      props = new Properties();

      props.load(new FileInputStream(f));
      String authips = props.getProperty("ip");

      if (authips != null) {
            addressChecker = new AddressChecker(authips);
         }
       mechanism = props.getProperty("mechanism");

      log.debug("Initialised DB_Authenticator");
}
```

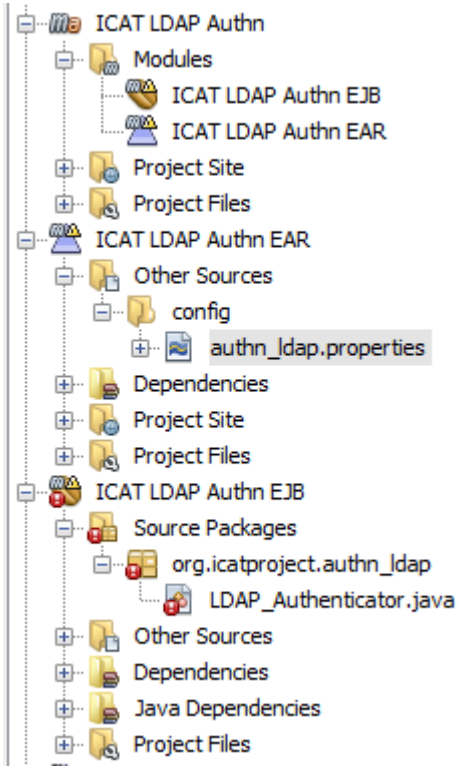
Example – ‘db’

```
@Override
   public Authentication authenticate(Map<String, String> credentials, String
          remoteAddr)  throws IcatException {

      if (addressChecker != null) {
         if (!addressChecker.check(remoteAddr)) {
            throw new IcatException(IcatException.IcatExceptionType.SESSION,
                  "authn_db does not allow log in from your IP address " +
                    remoteAddr);
         }
      }

      String username = credentials.get("username");

      String password = credentials.get("password");

      Passwd passwd = this.manager.find(Passwd.class, username);

      if (!passwd.getEncodedPassword().equals(password)) {
         throw new IcatException(IcatException.IcatExceptionType.SESSION,
               "The username and password do not match");
      }
      log.info(username + " logged in succesfully");
      return new Authentication(username, mechanism);
   }
```

Science & Technology Facilities Council

Example – ‘LDAP’

Simple classes, packages as JAR then EAR



Science & Technology Facilities Council

Example – ‘LDAP’

```
@PostConstruct
  private void init() {
      File f = new File("authn_ldap.properties");
      Properties props = null;
      props = new Properties();
      props.load(new FileInputStream(f));

      //same IP address checker stuff as before

      String providerUrl = props.getProperty("provider_url");
      if (providerUrl == null) {
         String msg = "provider_url not defined in " + f.getAbsolutePath();
         log.fatal(msg);
         throw new IllegalStateException(msg);
      }
      String securityPrincipal = props.getProperty("security_principal");
      if (securityPrincipal == null) {
         String msg = "security_principal not defined in " + f.getAbsolutePath();
         log.fatal(msg);
         throw new IllegalStateException(msg);
      }
          ...
          ...
```

Science & Technology Facilities Council

Example – ‘LDAP’

```
            ...
            ...

        if (securityPrincipal.indexOf('%') < 0) {
            String msg = "security_principal value must include a % to be
substituted by the user name "
                    + f.getAbsolutePath();
            log.fatal(msg);
            throw new IllegalStateException(msg);
        }
        this.providerUrl = providerUrl;
        this.securityPrincipal = securityPrincipal;

        // Note that the mechanism is optional
        mechanism = props.getProperty("mechanism");

        log.debug("Initialised LDAP_Authenticator");
    }
```

Science & Technology Facilities Council

Example – 'LDAP'

```
@Override
   public Authentication authenticate(Map<String, String> credentials, String
          remoteAddr)      throws IcatException {

      if (addressChecker != null) {
         if (!addressChecker.check(remoteAddr)) {
            throw new IcatException(IcatException.IcatExceptionType.SESSION,
                  "authn_db does not allow log in from your IP address " +
remoteAddr);
         }
      }

      String username = credentials.get("username");

      String password = credentials.get("password");

      log.info("Checking username/password with ldap server");
```

Example – 'LDAP'

```
    Hashtable<Object, Object> authEnv = new Hashtable<Object, Object>();
    authEnv.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.sun.jndi.ldap.LdapCtxFactory");
    authEnv.put(Context.PROVIDER_URL, providerUrl);
    authEnv.put(Context.SECURITY_AUTHENTICATION, "simple");
    authEnv.put(Context.SECURITY_PRINCIPAL, securityPrincipal.replace("%",
        username));
    authEnv.put(Context.SECURITY_CREDENTIALS, password);

    try {
       new InitialDirContext(authEnv);
    } catch (NamingException e) {
       throw new IcatException(IcatException.IcatExceptionType.SESSION,
             "The username and password do not match");
    }
    log.info(username + " logged in succesfully");
    return new Authentication(username, mechanism);

}
```

Science & Technology Facilities Council

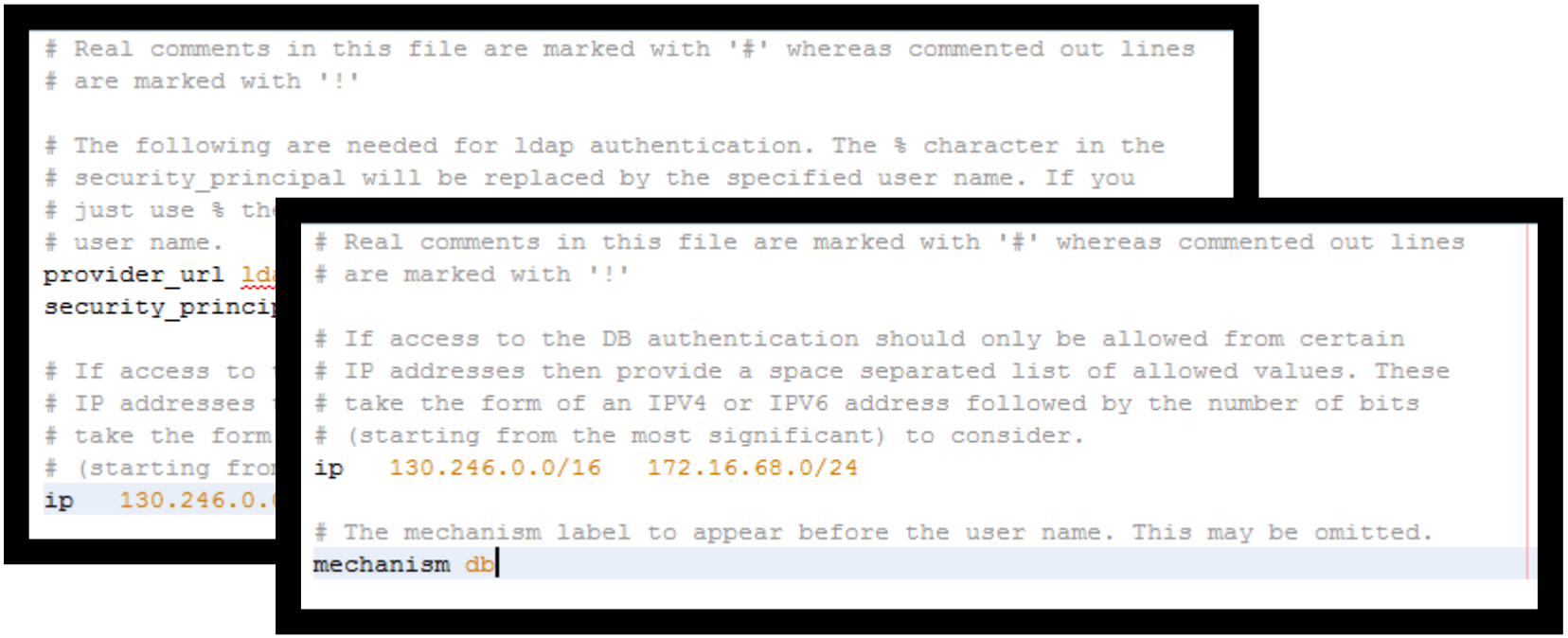
Example – 'UO Web service'

```
@Override
   public Authentication authenticate(Map<String, String> credentials, String
          remoteAddr)  throws IcatException {

      if (addressChecker != null) {
         if (!addressChecker.check(remoteAddr)) {
            throw new IcatException(IcatException.IcatExceptionType.SESSION,
                  "authn_db does not allow log in from your IP address " +
                    remoteAddr);
         }
      }
      String username = credentials.get("username");
      String password = credentials.get("password");
      try{
           String sessionId  = uoPort.login(username, password);
      }
      catch (Exception e) {
           thrown new IcatException(IcatException.IcatExceptionType.SESSION,
               "The username and password do not match");
      }
      if (sessionId == null {
         throw new IcatException(IcatException.IcatExceptionType.SESSION,
               "The username and password do not match");
      }
      log.info(username + " logged in succesfully");
      return new Authentication(username, mechanism);
   }
```

Science & Technology Facilities Council

Configuration

- Each mechanism can define it's own config file



Science & Technology Facilities Council

Configuration - ICAT

```
# Real comments in this file are marked with '#' whereas commented out lines
# are marked with '!'

# The lifetime of a session
lifetimeMinutes 120

# Provide CRUD access to authz tables
rootUserNames root

# Desired authentication plugin mnemonics
authn.list db ldap

# JNDI for each plugin
authn.db.jndi  java:global/authn_db.ear-1.0.0/authn_db.ejb-1.0.0/DB_Authenticator
authn.ldap.jndi java:global/authn_ldap.ear-1.0.0/authn_ldap.ejb-1.0.0/LDAP_Authenticator
```

Science & Technology
Facilities Council

Questions?

www.icatproject.org
code.google.com/p/icatproject

http://groups.google.com/group/icat-developers/
http://groups.google.com/group/icatgroup/

icat-developers@googlegroups.com
icatgroup@googlegroups.com